A Survey of Next-Generation Reconfigurable Architectures for Embedded Computing

Raj Krishnamurthy
{rk@cc.gatech.edu}

1.0 Introduction

The capabilities of an ASIC(Application Specific Integrated Circuit) cannot be changed once it has been produced or 'taped-out', on the other hand, a processor allows more flexibility by providing an ISA(Instruction-Set Architecture) and allowing programs to change sequences of instructions within the confines of a memory consistency model. Software allows increased flexibility as it can be written in a High-level language and compiled into a processor ISA instruction sequence. An FPGA(Field Programmable Gate Array) so called because it can be programmed in-situ lies 'in between' hardware and software. The intent of an FPGA design is to provide a designer with the flexibility of software at hardware speeds. FPGAs have been developed with anti-fuse technology(one-time programming) but the most popular are SRAM based versions that may be programmed an infinite number of times.

The notion of Reconfigurable Computing in this paper is related to FPGA architectures(which may be standalone) and also to architectures that use FPGAs as one of their components. The capabilities of a system are significantly enhanced when a heterogeneous architecture comprised of different components and FPGAs are used in an integrated fashion.

FPGAs have been popular for logic prototyping, logic emulation and 'glue logic' and are being increasingly used for hardware acceleration of application-specific computations not limited to image processing kernels and video streams. Rapid prototyping and their use as a cheap ASIC replacement are allowing penetration into newer market segments.

This report starts with this introduction in Section 1, the need for Reconfigurable architectures is described in Section 2, Section 3 delineates the criteria for selection of candidate architectures, Section 4 describes academic research projects, industry research projects are described in Section 5 with Section 6 describing key common attributes, Section 7 describes the shape of things to come with Section 8 concluding the paper.

2.0 Why Reconfigurable Architectures for Embedded Computing?

For analyzing the suitabilty of Reconfigurable architectures for Ubiquitous computing and embedded computing applications, it is important to delineate the characteristics of applications found in this space.

Ubiquitous Computing applications are characterized by the following properties -

- Balance user input with system operation and performance
- Embedded low-power operation
- Workloads may be highly heterogeneous with a mix of control-dominated and data-dominated workloads
- Workloads frequently involve media streams or other computation that are stream-oriented
- Real-time predictable response for systems used in mission-critical or life-critical environments
- Processing of sensor data and positioning is increasingly data-centric due to convergence of computation, communication, sensing and actuation
- Development requires small time-to-market
- Rapid prototyping and logic emulation are important for system construction

2.1 Strengths of Current FPGA/Reconfigurable Architectures

Reconfigurable architectures have a number of strengths that make them suitable for development and use in Ubiquitous Computing systems [HauckReport]-

- Rapid prototyping, logic emulation and cheap ASIC replacement
- Extreme parallelism extraction potential with ability to vary bit-widths(compared to fixed width of a processor) ie use right computational power for a given application. It is an overkill to engage a processor to operate on three bit numbers. Allows power-savings.
- Deeply customized spatial and temporal pipelining for operation of multiple contexts
- Partially evaluated circuits that can be collapsed when for example constant operators are used
- Hardware virtualization allowing multiple contexts and run-time partial reconfiguration of contexts
- late-binding during execution with required components
- Ability to uncover data parallelism in streams
- Mixing with other components allows multi-granular operation - coarse grained versus fine-grained operation

2.2 Issues in Current Reconfigurable Architectures : Challenges for Next-Generation Architectures?

A successful Next-Generation architecture will address the needs of Ubiquitous Computing applications specified earlier and also address some of the weaknesses described here[HauckReport] -

- Poor floating-point performance. This can be resolved by integration with a processor core optimized for floating-point calculations[UCGarp].
- Limited resources. This is being addressed by Million-gate FPGAs like Virtex[XilinxWeb]
- Poor data proximity. Vendors are integrating memory(CAM and RAM) in next-generation

architectures[AlteraWeb]
- Poor power efficiency. A Xilinx 4003 part can consume 2 W at 20MHz[XilinxWeb].
- Fine granularity architectures unsuitable for datapath style coarse-grain multimedia computations. FPSC with mixed fine - and coarse-grained structures may be suitable?[HauckFuture, CMU99].
- 90% of chip area is interconnect and interconnect utilization can severely degrade power-efficiency.
- High Configuration times and compilation times. Xilinx Virtex and Chimaera are addressing this[XilinxWeb, NWU]

3.0 Criteria for Selection of Candidate Architectures

Six research projects and three industry products/projects were selected for this report. The criteria for selection was based on -

- Relevance to Embedded computing/Ubiquitous Computing/Media Processing space ie low power and media-oriented
- Represent different approaches - standalone versus Systems-on-a-Chip
- Resolve issues in current architectures
- Push frontiers of existing technology - run-time reconfiguration and computational density

The final candidates were grouped as follows -

Table 1. Reconfigurable Architecture Candidates

Academia/Industry?	**Project name**	**Location**
University	Pleiades	UC Berkeley
	Piperench	Carnegie Mellon University
	DPGA	MIT
	BRASS	UC Berkeley
	Chimaera	University of Washington and Northwestern University
	Morphosys	UC Irvine
Industry	Virtex, Spartan, Coolrunner	Xilinx Corp.
	Apex, Ace	Altera Corp.
	Orca	Lucent Corp.

4.0 Description of Academic Research Projects

4.1 Pleiades

Pleiadas is an effort at UC Berkeley developing Ultra-Low-Power, hybrid and reconfigurable computing architectures in the embedded/ubiquitous computing space. The effort is based on a number of novel techniques[UC96] -

- The approach allows mixed-granularity elements on a single die - an embedded processor, configurable elements(programmable at the gate, datapath or functional level). The embedded processor provides support for execution of control dominated application kernels. Other units provide support for data-dominated streams such as those seen in traditional DSP kernels, video and audio streams.
- A template for building domain-specific computation blocks. For a given application, the number of processors and configurable units may be specified based on the computational and power requirements of the application.
- A reconfigurable interconnect has been developed for low-power operation with low-swing characteristics. This uses techniques in asynchronous or self-timed signaling. The technique to reduce interconnect power is based on a locally synchronous and globally asynchronous model which allows for adaptive voltage and clock scaling.
- The embedded core processor uses performance-adaptive voltage scaling.
- Development of a low-power FPGA module which uses - small granularity cells, hierarchical interconnect and reduced voltage swings. *This has led to dramatic improvements in power savings over commerically available products of around ~x40.*

A number of chips have been built as a result of this effort [UCplweb] -

- Maia - A prototype Pleiades processor was constructed and testing was underway during September 1999. The processor combines an embedded ARM-8 core with 21 satellite processors(1 low-power FPGA, 2 Multiply-Accumulate units, 8 SRAMs, 2 ALUs and 8 address-generators). The units are connected by a two-level hierarchical reconfigurable inteconnect. The processor is intended for nominal operation at 1V and at 40MHz with an area of 4.5x6 sq.mm and with around 1.2 million transistors. The projected power consumption for a complete VCELP voice coder is below 1mW.
- A separate low-power FPGA module. It consists of an 8x8 array module(2x4 LUT) with a mixed 0.8-1.5 V supply and is clocked at 125 MHz. An 8-bit adder built using this low-power FPGA has a delay of 20nsec and consumes 70 times less energy than its comparable Xilinx implementation.
- A set of four chips were also built using dynamic voltage scaling. These include an ARM8 core with integrated peripherals, a memory controller, a DC-DC converter, and a low-energy voltage-scalable SRAM. The processor can operate between 8 and 100 MIPS with the energy dissipation per instruction around 0.24nJ.

The next two sections describe the low-power FPGA and MAIA processor as these are most pertinent to the embedded/Ubiquitous computing space.

4.1.1 Pleiades Low-Power FPGA[UC99]

FPGAs are being used in the embedded computing/ubiquitous computing space as performance accelerators beyond the traditional uses of logic emulation and "glue" logic. Commercial FPGAs suffer from high power consumption in the order of Watts even at low clock-frequencies. For the use of FPGAs in low-power environments like cell-phones where power consumption is of the order of mW, a

new FPGA design is needed to address low-power environments. In a traditional FPGA structure, interconnect dominates the area(upto 90%) and speed performance(because of fine-programmability of an FPGA). The energy breakdown of a Xilinx 4003 A part is shown in Table 2 [UC99].

Table 2. Energy Breakdown of Xilinx 4003A

Interconnect	65%
Logic	5%
IO	9%
Clock	21%

As interconnect power consumption is dominant in the breakdown , the focus of the Pleiades FPGA architecture is on the interconnect.

Design Approach

The approach is based on optimizations at the architectural and circuit-levels. This is done concurrently so that circuit-level parameters can be used to update the architectural model. The approach involves the following components - interconnect, CLB architecture, clock distribution with circuit-level optimizations[UC99].

Interconnect Architecture

The interconnect on an FPGA serves to provide connectivity between CLBs or Configurable Logic Blocks. The interconnect on the Pleiades FPGA is based on three levels[UC99] -

- Level 0 - Nearest Neighbor
  The project found that is was optimal for a CLB to connect to 8 neighbouring CLBs with a fan-out of 8. Compared to the traditional mesh connection, it was found that this gives an improvement in the ED(Energy-Delay) product of around ~x3.

- Level 1 - Mesh Interconnect
  This level of interconnect is similar to that found in the Xilinx architecure - inter CLB communication is through C-box or Connection Box. Horizontal and Vertical routing direction change is done at the S-Box or Switch Box. This level is provided for communication beyond level-0 NN connections.

- Level2 - Hierarchical Interconnect
  In the case of the Mesh interconnect, delay increases as l*l and Energy-Delay increases as l*l*l. The level-3 interconnect addresses this problem using a binary-tree interconnect for longer wires. Shorter wires are still routed using the Mesh interconnect. Also, clustering optimization has been used by inverting the conventional cluster found in binary tree architectures. Since the mesh architecture is being used for short-haul connections, inverting the cluster allows longer connections (in the original cluster) to be accessible as shorter connections(at the same lowest level).

CLB Architecture

From Table 2. above, contribution of the CLB or logic to the Energy total is minimal. Because of the fine-grained nature of the CLB, effects on interconnect utilization or interconnect energy may be pronounced. The CLB structure consists of 4 x 3-input lookup tables(LUTs) which have been found to be optimal from an energy perspective[UC99].

Clock Distribution

From the energy table above, the contribution of the clock to the energy total is around 20% and the energy concentration is in the global distribution network. The design uses Dual Edge Triggered FFs to bring down the activity in the clock distribution network by almost a factor fo two. The authors argue that the increase in complexity because of this change is minimal, with 3FFs/CLB, the contribution in area from the flip-flops is ~0.8%. Low-swing signaling has also been used to bring down the energy consumption further [UC99].

Summary of Circuit-Level Optimizations

There are basically two circuit-level optimizations used by this approach - one is sizing transistors based on Energy-Delay product in a typical path(please see [UC99]) and the other is using a low-swing circuit technique based on SDVST-II. The SDVST-II technique gives an improvement according to the authors of more than a factor of two.

Implementation

The low-power FPGA was implemented by the authors in a 8x8 array with size of 2mmx2mm in a 0.25U 6 metal CMOS process[UC99].

Results

*Preliminary data from the layout show an improvement of ~x40 over a Spartan Series Xilinx FPGA.*The Table 3 below compares a Xilinx FPGA, an Altera FPGA and the Pleiades FPGA. Power consumption of the Xilinx and Altera parts are of the order of Watts whereas the power consumption of the Pleiades FPGA is of the order of mW.

Measurement	Xilinx XC 4000L	Altera FLEX 10K	Pleiades FPGA
Single FF driving 9 segments(pJ)	320	485	4.85
1K Arrays of 16 bit counters at 30MHz(mW)	1.5E3	3.1E3	21.8
Maximum Topology Frequency(MHz)	166	100	62.5(125)

Table 3. Performance Results

4.1.2 MAIA: Reconfigurable Multi-Granular Processor

The MAIA reconfigurable processor is described here because it incorporates FPGAs in its structure. The description will focus on the place of the FPGA in the structure and the communication mechanisms to communicate with other units on the chip. The MAIA reconfigurable processor is based on a heterogeneous SOC(Systems-on-a-chip) style design with different units. A heterogenous-style design allows energy, speed and flexibilty to be met at the same time while allowing strengths of different units to be harnessed for a common design cause. The authors argue that this allows better algorithm/architecture matching than other designs. The base architecture consists of a control processor and other satellite units(can be processors, FPGAs or other units such as MAC). The model of computation and reconfiguration is as follows - A sequntial thread is instantiated on the control processor and "split" operations from the control processor "spawn" computations or activities on satellite units. On asynchronous completion of activities, satellite units may "join" with the main control processor. The architecture is reconfigurable in two respects - Inter-satellite communication is scheduled by the control processor and also between the control processor and the satellite units dynamically during run-time and also statically with compiler support. Second, the reconfigurable architecture may also have FPGAs that support run-time reconfiguration and fast context switching between configurations [UC2000].

Architecture

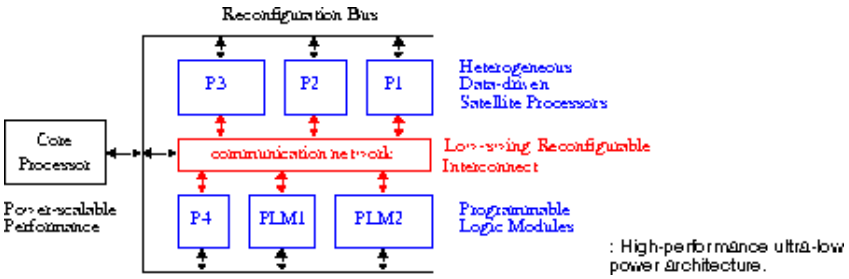


Figure 1. Architecture of the MAIA Reconfigurable Processor[UCplweb]

The basic architecture consists of the core processor, PLMs(Programmable Logic Modules or PLMs) and other units(MACs, address generators and memories). The MAIA processor consists of a microprocessor core(ARM8) and 21 satellite processors: two MACs, two ALUs, eight address generators, eight embedded memories(4 512x16bit, 4 1kx16bit) and an embedded low-energy FPGA. Connections between satellites are accomplished through 2-level hierarchical mesh-structured reconfigurable interconnect network. The ARM8 uses an interface control unit to configure and communicate data with satellites. The address generators and embedded memories are distributed to supply multiple parallel data streams to the computational elements.

Programmable Logic Module(PLM-FPGA)

The PLM-FPGA is similar to the Pleiades FPGA described above except that it is a 4x8 array of 5-input

3-output CLBs. All the architectural modifications described above have been included in the design, A low-swing circuit is used with 1.5V for the logic blocks and low-swing signal lines run at 0.8V[UC2000].

Communication Interface

Inter-satellite communication is realized using a 2-phase self-timed handshaking scheme with REQUEST and ACKNOWLEDGE signals. This is done in a globally asynchronous and locally synchronous fashion. This has the advantage of reducing power-consumption by ensuring that the module is only activated when data is ready. Data links are 16-bit fixed-width wide words with 2-bit control tokens[UC2000].

For communication between the ARM8 core and the satellite units, an interface control unit is used. The interface control unit is also used for loading configurations into the Programmable Logic Modules and configuring the programmable interconnect.

Reconfigurable Interconnect Architecture

Many DSP kernels contain a number of regular computations and the energy efficiency of the interconnect may be enhanced by preserving the locality in the computations. Interconnect like crossbars may provide a lot of flexibility but may not be all that energy efficient. Interconnects with regular structures are likely to be the most energy-efficient. A two-level hierarchical mesh has been developed for energy efficiency, preserving locality and enhanced algorithm/architecture matching. Short-haul interconnections are supported by the first layer with switchboxes for changing routing directions from horizontal to vertical. Long-haul interconnects are actually implemented on higher metal-layers to provide connectivity at the second-level. Hierarchical switchboxes are provided at key connection points to provide connectivity between first - and second - layer of the interconnect[UC2000].

Communication energy is further reduced by employing a low-swing 0.4V pseudo-differential signaling scheme. As an asynchronous clocking protocol is used, the clock signal is generated from the handshaking signals. The low-swing signaling reduces the interconnect energy by a factor of 3.4 compared to a full-swing CMOS implementation. The interface of the core processor module and the reconfigurable interconnect is shown in Figure 2.

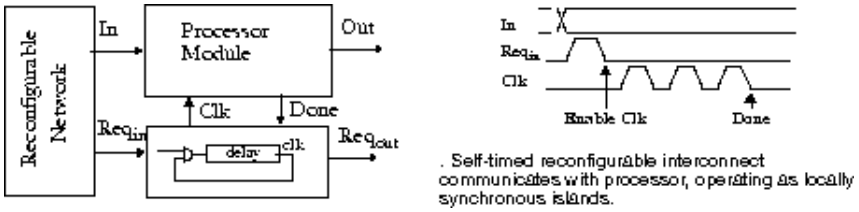


Figure 2. Self-timed Reconfigurable Interconnect[UC2000]

Implementation

The MAIA chip was implemented using 0.25U 6-level metal CMOS process with a supply voltage of 1V and additional voltages of 0.4V and 1.5V, The die size of the implementation was 5.2mmx6.7mm with 1.2 million transistors at 40MHz with an average power dissipation of 1.5-2 mW[UC2000].

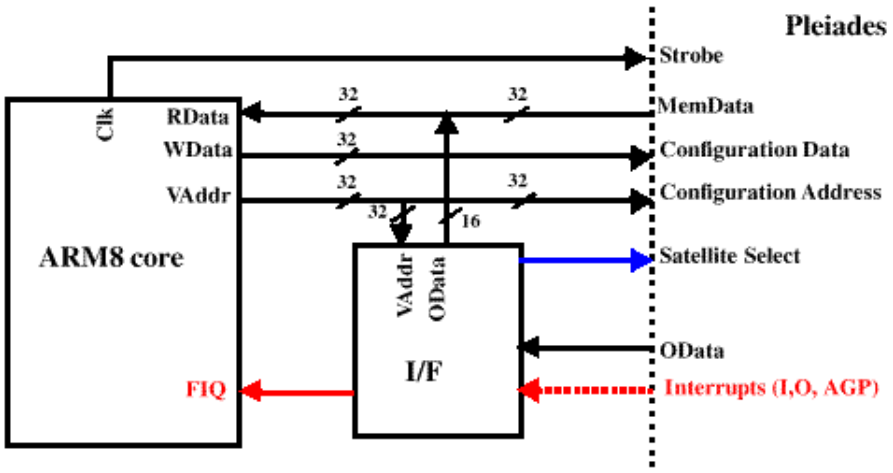


Figure 3: Interface of ARM8 core with satellite units[UCplweb]

Performance

The performance of different modules on the chip is shown in the Table 4 below:

Hardware Modules	Pipeline Speed(ns)	Energy Consumption per Operation(pJ)	Area(mm*mm)
MAC	24	21	0.25
ALU	20	8	0.09
Memory(1Kx16)	14	8	0.32
Memory(512x16)	11	7	0.16
Address Generator	20	6	0.12
Interconnect Network	10	1	NA
FPGA	25	18	2.76

Table 4. Performance of Hardware Modules

4.1.3 Comments

The objectives of this project completely satisfy the requirements set forth earlier. The ability to build low-power FPGAs and integrate them with multi-granular SoC architectures will allow the flexibility and power/performance tradeoffs. The issues here will be in engineering for development of suitable solutions keeping in pace with process technologies.

4.2 Piperench

Piperench is an effort at CMU led by Seth Copen Goldstein looking at developing Reconfigurable Coprocessors for streaming multimedia applications. Streaming applications usually have the following characteristics - different bit-widths from main processor(usually 16bits), data dependencies allow parallel execution of units, the implementation can usually be pipelined, constant propogation can usually be performed that allow a generic multiplication to be transformed into a constant multiplication which can be implemented as shifts(for example), the input values are also reused many times. Commercial FPGAs according to the authors have the following limitations -

- With fixed fine logic-granularity, random logic can be implemented right and not differential datapath operations as in multimedia computations.
- Configuration times are of the order of microseconds to hundreds of milliseconds.
- FPGAs require redesign or recompilation to benefit from future generations of the chip.
- As kernels of a fixed and small size can only be implemented, this leads to poor algorithm/architecture matching.
- Also, compilation times are very high.

Piperench strives to alleviate the above problems by using the notion of hardware virtualization where physical pipeline stages are configured to accommodate virtual pipelines or stripes by overlapping configuration and execution. As streaming computations can be represented as stages of a pipeline, they can be mapped to physical pipeline stages(made up of Piperench PEs) by overlapping configuration of the physical pipeline stages with execution of the stages. Also a key contribution of this architecture is run-time reconfiguration and partial run-time reconfiguration of pipeline stages[CMU99, CMU98, CMU97, CMUweb].

Architecture

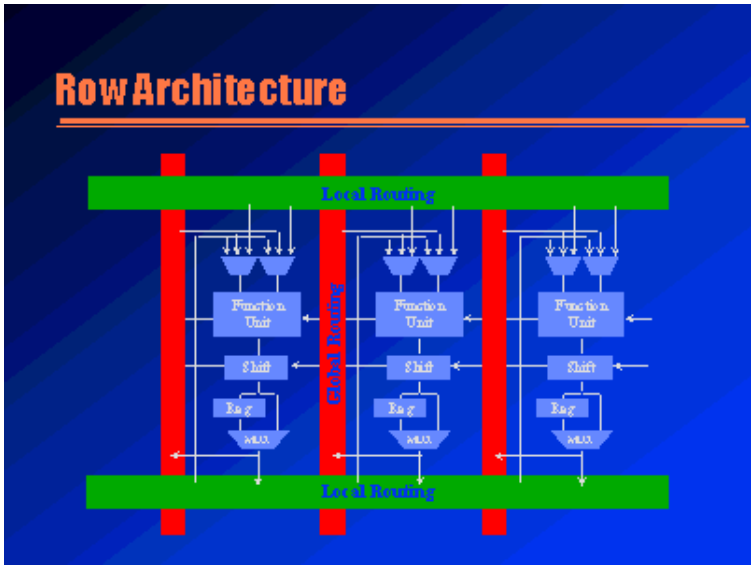


Figure 4. PE (Processing Element) Architecture in Piperench[CMUweb]

Each PE in Piperench basically consists of a interfaces to global interconnect and local interconnect. Inter-stripe communication is provided by Pass registers. Combinational logic is implemented using a set of N B-bit wide ALUs. The carry lines of PipeRench's ALUs may be cascaded to construct wider

ALUs. A physical stripe is implemented as an array of PEs[CMU99].

Implementation

During the Summer of 1999, the project was considering a design in 100 mm*mm of Silicon in a 0.25 U process. Half of the area is for reconfigurable fabric and the other half is for memory to store virtual stripes, control and chip I/O[CMU99].

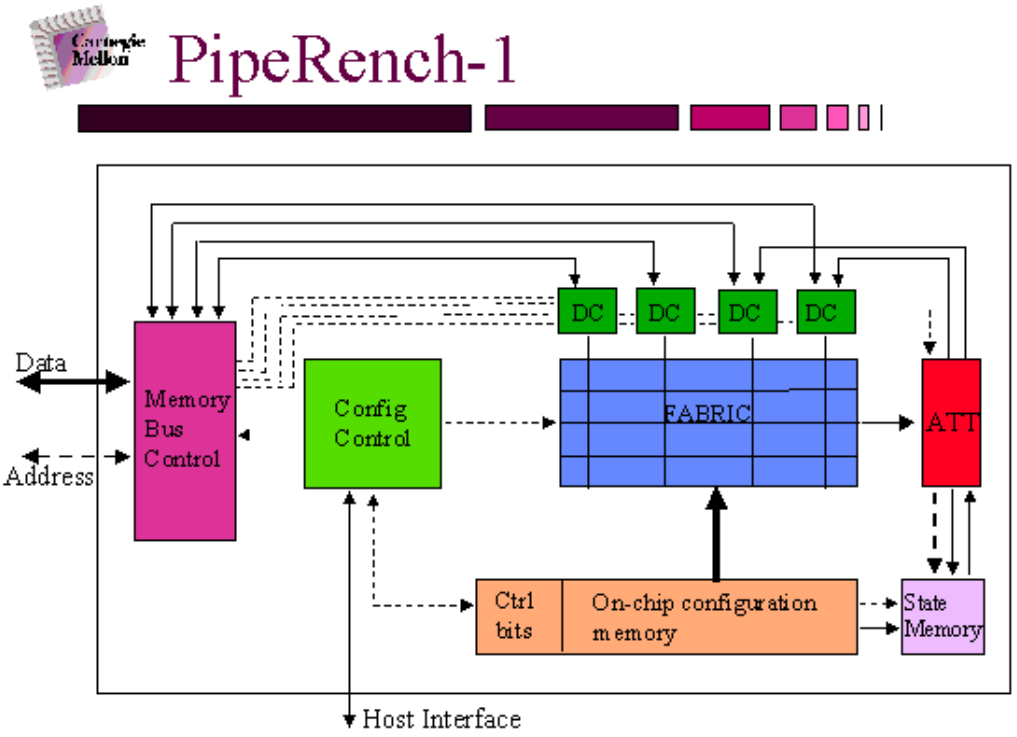


Figure 5. Piperench Architectural Layout[CMUweb]

Performance[CMU99]

Performance for certain computing kernels were found to be ~x190 times that on a modern RISC processor.

Application/Kernel	Speedup
ATR(Automatic Target Recognition)	189.7
Cordic	15.5
DCT	11.3
DCT-2D	11.3
FIR	63.3
IDEA	42.4
Nqueens	26.0
Over	57.1

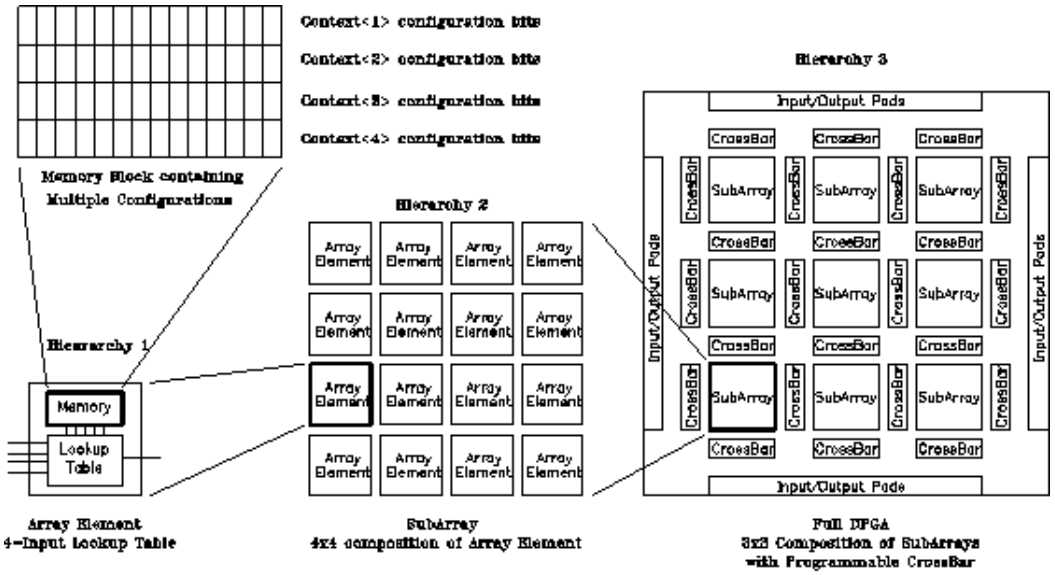
Table 5: Results

4.2.1 Comments

The project does not directly address the issue of low-power but addresses the idea of reconfigurable computing for streaming multimedia applications found in many ubiquitous applications. Can low-power optimizations be applied to this architecture?. The techniques used in [UC99] may be applied to this work. The interconnect for Piperench may need to be changed as it is currently a cross-bar. So suitable power/performance tradeoffs must be made in the Energy-Delay product space. A non-trivial engineering task but, the principles of hardware virtualization and the study of application charactersitics may be used by other projects for development of low-power reconfigurable Ubiquitous Computing applications.

4.3 DPGA

DPGAs are Dynamically Programmable Gate Arrays and are basically multi-context FPGAs. The primary motivation is that FPGAs allow spatial processing of dataflow kernels but suffer from poor temporal utilization. As soon as the computation is complete on a particular CLB, the next data set may be pipelined. DPGAs provide both spatial and aggressive temporal pipelining at the level of a CLB(the basic block) The first DPGAs were prototyped at MIT as part of the Transit project by Andre DeHon.

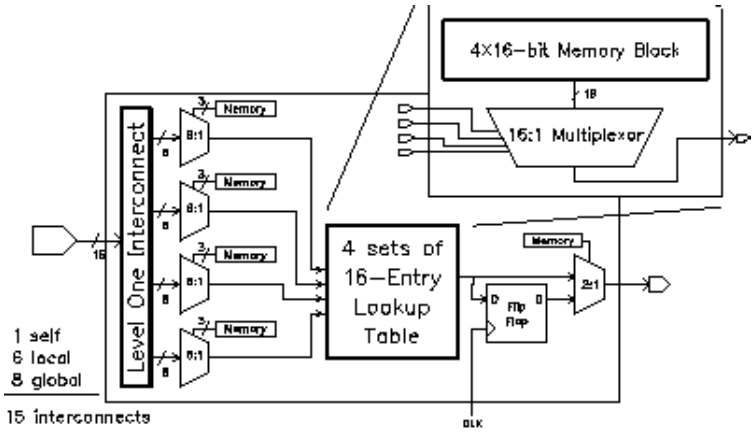


Architecture and Composition of DPGA

Figure 6. Architecture of a DPGA[DPGAweb]

The architecture of the DPGA looks like a SIMD array with crossbars providing the interconnect. For the prototype developed in 1994, a four context DPGA was developed with DRAM used as configuration cells, non-intrusive background loading was provided with automatic refresh of dynamic memory elements. A wide-bus architecture allowed high-speed context loading with interconnects at the local and global level [DPGA, DPGAweb].

Each array element in the array looks like as shown in Figure 7. below.

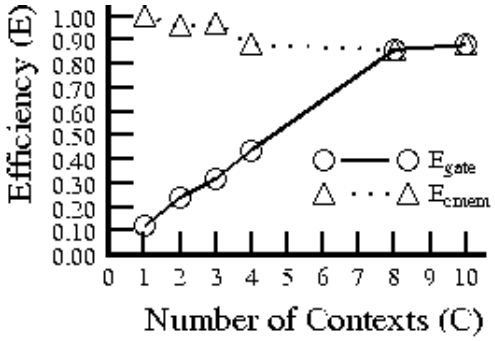


Simplified Block Diagram of the Array Element

Figure 7. Array Element[DPGA]

There are four DRAM cells to store the configuration contexts, 4 sets of 16-entry LUTs for generation of random logic and a flip-flop for state-storage.

Performance Results



DES Benchmark Utilization versus Number of Contexts
Figure 8. DES Benchmark Utilization[DPGA]

The utilization for DES benchmarks are impressive. Utilization of almost 90% is seen with eight contexts[DPGA, DPGAweb].

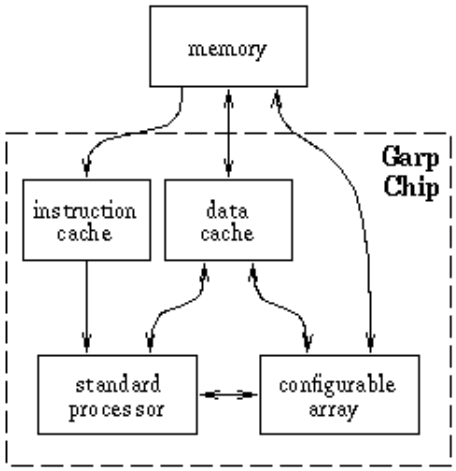
4.3.1 Comments

DPGAs are interesting because they allows enhanced real-estate utilization of FPGA capacity using spatial and temporal pipelining. It also uses two levels of interconnect. The rapid switching of context can aggravate the power problem in a Reconfigurable architecture used in a Ubiquitous Computing application. Multi-context FPGAs are generally interesting because they allow the hardware to be treated as "virtual hardware" with rapid switching of multiple context as in an Operating Systems scheduler. Multiple context switching coupled with run-time reconfiguration can be very powerful for increased performance in reconfigurable architectures.

4.4 BRASS : GARP and SCORE

Garp and SCORE (Stream Computation Oriented for Reconfigurable Execution) are two projects at UC Berkeley led by John Wawrzynek. Garp was interesting because it was one of the first projects to identify one of the weaknesses of FPGAs - inability to perform loops and control well and to recognize the fact that FPGAs were mostly suited for dataflow computations.

Garp allows integration of a RISC core(suitable for control and tight loops) to co-exist with an FPGA module. This allows a "hybrid" architecture that can be used to accelerate performance for dataflow computations while allowing standard application kernels with control and tight loops to be run as usual.



Garp Block Diagram

Figure 9. Garp Architecture[UCGarp]

The standard processor in the Figure above is single issue RISC core MIPS-II. The gate array consists of 32 rows by 23 columns of logic blocks with the 24 th column used for communication outside the array. Each CLB or logic block consists of 2-bit input and produces two-bit outputs. Four bits of data state may be included on each logic block for a total of 92 bits per row. By use of row increments, partial configuration of the array is possible. Reconfiguration time from external memory takes about 12 external bus cycles per row. Reconfiguration time from the integrated cache is 4 cycles(independent of the number of rows). The RC array has the ability to perform data caches or memory accesses independent of the MIPS core. Virtual memory, supervisor mode and protected execution of multiple processes is supported by the GARP architecture with control synchronization provided between the MIPS core and the RC array[UCGarp].

Performance results from the GARP architecture, compare the GARP chip with a Sun Ultrasparc 1/170[UCGarp]

Table 6. Performance Results

Benchmark	**167 MHz SPARC**	**133 MHz Garp**	**Ratio**
DES Encrypt of 1MB	3.60 s	0.15 s	24
Dither of 640x480 image	160 ms	17 ms	9.4
Sort of 1 million records	1.44 s	0.67 s	2.1

4.4.1 SCORE

SCORE is based on the observation that Reconfigurable hardware architectures lack a clear architectural and programming model. The SCORE project attempts to provide a stream-oriented computational model to address the issues described above. This is very pertinent to the Embedded / Ubiquitous computing space as most applications in this space are media streams or streams like those seen in DSP applications. Computations are broken into compute pages and are the basic unit of virtualization and reconfiguration. They are managed just like the VM system in modern operating systems. Compute pages are linked together with streams. Configurable Memory Blocks are distributed in the array and are used to hold - configuration and state for virtual compute pages, stream buffers and segments for virtual memory space. The key point is that pages may be created and destroyed dynamically and may be used with I/O channels for input/output. Also, an OS manager is used to manage pages much like a VM manager in an Operating System[UCScore].

A spatial implementation is shown in Figure 10 below with components of a video processing application[UCScore].

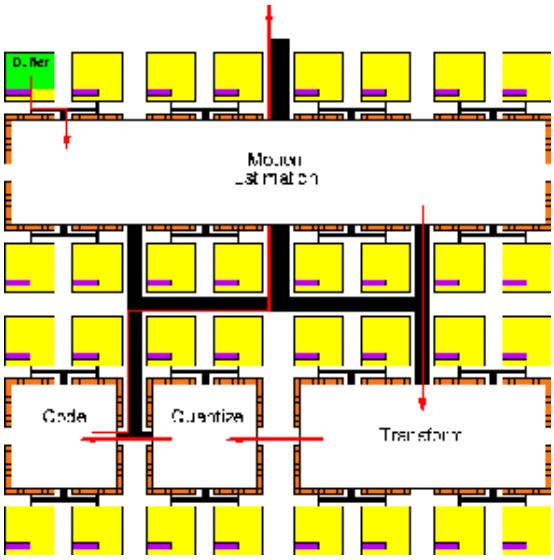


Figure 10. Spatial Implementation[UCscore]

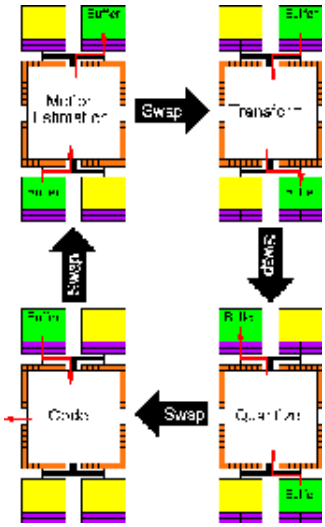


Figure 11. "Switched Implementation"[UCscore]

The "Switched Implementation" in Figure 11 uses the buffers for stream buffers for communication between stages in a pipeline.

Preliminary performance results from SCORE are shown in the Figure below

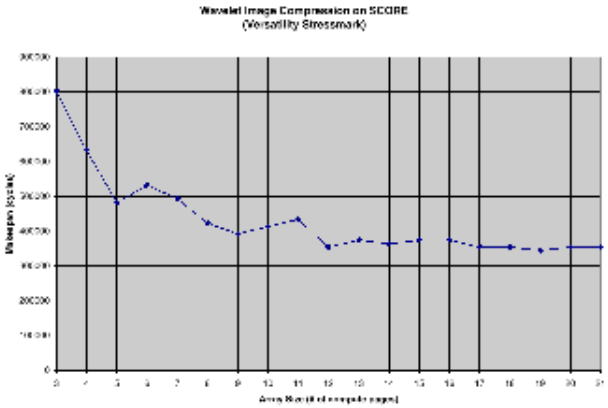


Figure 12. Wavelet Performance[UCscore]

With the wavelet transform design, the number of compute pages are varied from 3 to 21 where the asymptote of full spatial performance is seen. At 5 compute pages, the speedup overhead is minimum for each additional page added. It is important to note that at 12 physical compute pages, spatial asymptote performance is seen[UCScore].

4.4.2 Observations

GARP is interesting because it was one of the first projects to propose a hybrid architecture to integrate control processing using a RISC processor and data-dominated processing using a RC array or an FPGA. SCORE is relevant to Ubiquitous computing because it provides a model for stream computations seen very often in sensor applications and other media-oriented Ubiquitous computing workloads and hardware architectures to deal with these workloads.

4.5 Chimaera and RTR

Chimaera is similar to GARP[UCGarp]. It tightly couples a RISC processor core and a reconfigurable array as shown in Figure 13 below. It is an effort at Northwestern University led by P. Bannerjee and Scott Hauck.

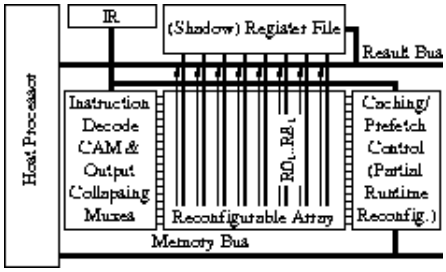


Figure 13. Chimaera RPU[NWUweb]

The architecture of the reconfigurable computing array is shown in Figure 14 below.

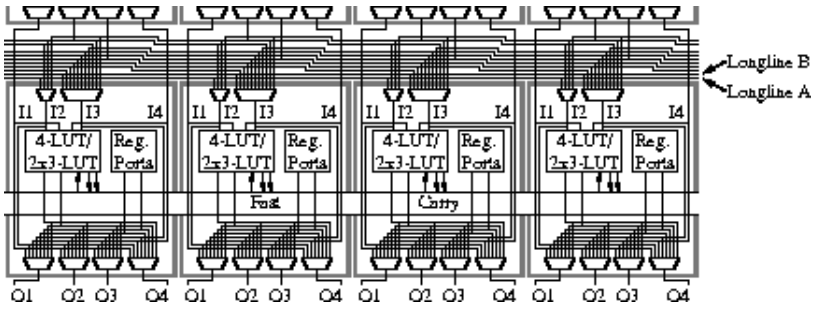


Figure 14. Chimaera Reconfigurable Array Architecture[NWUweb]

The interesting aspects of the RC array are the high-performance carry chains which allow high-speed adder designs[NWU].

The key contributions of the Chimaera are beyond the architecture described above. They lie in the two key areas below -

- Configuration management - The ability to allow multi-context designs to exist with minimum context switching overhead. The project has developed strategies to allow configuration compression and on-chip storage of configurations. Strategies for high-speed transfer of configurations, configuration caching, dynamic contexts and configuration prefetching have also been developed. The architecture also supports partial run-time reconfiguration which allows a part of the design to be updated without a context "switch-out" of the whole configuration[NWUweb].
- High-level Compilation tools - Allow compilation of high-level source for execution on the host

processor / RC array configuration[NWUweb].

4.5.1 Comments

Chimaera is interesting because of the multiple configuration planes and the ability to support multiple contexts. Configuration caching and configuration compression strategies coupled with partial run-time reconfiguration make this effort very interesting. Run-time reconfiguration can be an important strength for Ubiquitous computing applications, applets or agents can be transferred from appliances or devices and the computations on these may be accelerated using the RC array by partial run-time reconfiguration
without maybe even stalling the processor.

4.6 Morphosys

Morphosys is an effort at UC Irvine led by Nader Bagherzadeh. This architecture is interesting because to combines a number of earlier research ideas described above - Dynamic programmability from DPGAs(multi-context), tight coupling with main processor (Garp) and SIMD like arrangement as in the DPGA prototype to enable dataflow style computations[DPGA, UCGarp, NWU]. The difference from DPGA is that this allows coarser-grained structures as part of the logic block. The floorplan of the Morphosys chip is shown below [UCI]-

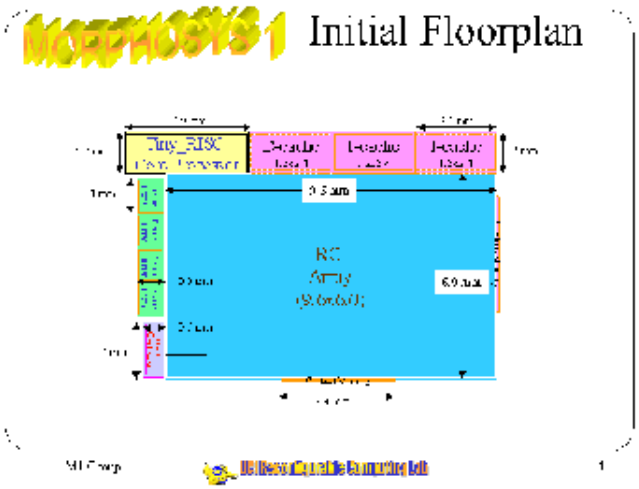


Figure 15. Morphosys Floorplan[UCIweb]

Key features are operations on 8/16-bit data, context words which specify an instruction opcode for the RC. 32 contexts may be stored in multiple configuration planes. Dynamic reconfiguration is supported without interrupting RC operation. Control processor and the RC array are on the same die and interface to memory is through a fast DMA controller.

Architecture is shown below -

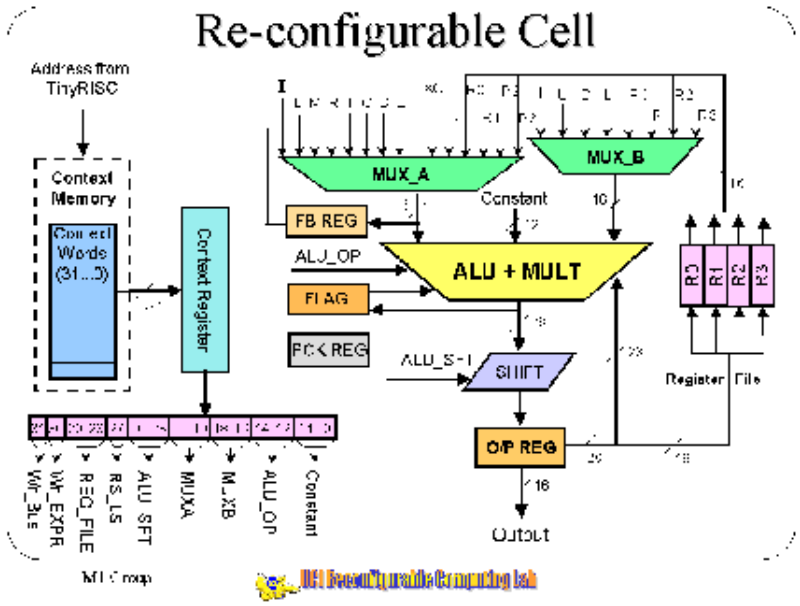
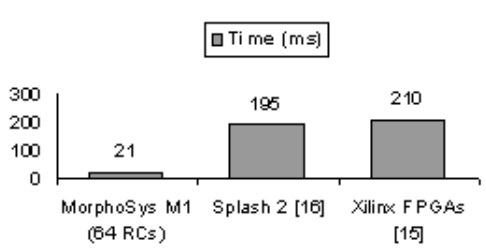


Figure 16. RC Array architecture[UCIweb]

The RC array architecture shows context memory and the coarse-grain ALU and MULT for dataflow-style computation mapping.

The performance for ATR(Automatic Target Recognition) applications is shown below -



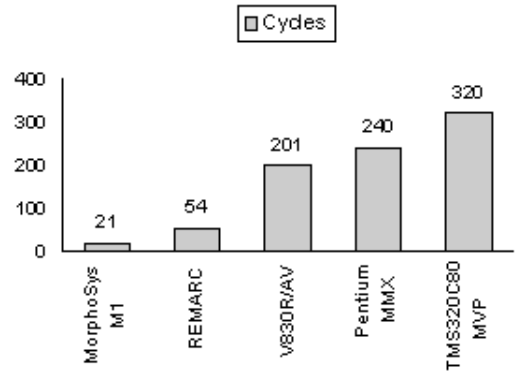


Figure 17(1). ATR Performance[UCIweb] **Figure 17(2). DCT Performance[UCIweb]**

DCT performance is also shown in Figure 17(2) which forms the basis for many video coding applications including MPEG compression[UCIweb].

4.6.1 Comments

Morphosys is interesting because it brings ideas from previous architectures like dynamic programmability and SIMD style arrangement and retargets it for a croaser-grained implementation similar to datapath style arrangements. The coarse-grained arrangement allows it to process multimedia

streams and other dataflow computations with speedups shown above.

5.0 Description of Products and Projects in Industry

This section describes some of the projects from some of the key players in industry. Some projects have been realized into actual products and are described below. Other projects are still in the fabrication phase and are duly noted where so. The following description will describe the state-of-the-art with future product announcements discussed where vendors have released such information.

5.1 Xilinx

Xilinx6200

Xilinx XC6200 was one of the most influential architectures after the current generation of architectures which included the Xilinx 4000 series. *An important feature of the Xilinx XC6200 was that it allowed a partial reconfiguration of the FPGA during run-time*. Earlier, the whole unit had to be reconfigured taking milliseconds to load a new configuration. This of course leads to an increased amount of routing structure(called FastMap). The FastMap allows an external processor or agent to read or write a register or logic block on the device on-demand[XilinxWeb].

XilinxVIRTEX

The new Virtex family from Xilinx support gates from *50k to 4 million gates*. The Virtex part can be clocked at 200MHz and are compatible with 64-bit/66 MHz PCI. The family allows for multi-standard I/O devices and connects directly to ZBTRAM devices. The clock-management circuitry is enhanced with four dedicated delay-locked loops and four primary low-skew global clock distribution nets. The memory system is hierarchical with support for 16-bit, 32-bit and 16-bit dual ported RAM. Each logic block has a dedicated carry logic for high-speed arithmetic, dedicated multiplier support, cascaded chain for wide-input functions. Abundant latches exist with clock enable and dual sync/async set/reset. In-system configuration is SRAM based with unlimited reporgrammability and supports four programming modes. The Virtex-E family is built on a six-layer 0.18U CMOS process. Like the XC6200 family, *Virtex allows run-time partial reconfiguration.*

Architecturally, the Virtex array consists of CLBs(Configurable Logic Blocks) and IOBs(I/O Blocks). CLBs provide the functional elements for constructing logic. IOBs provide the interface between package pins and CLBs. CLBs interconnect through a general routing matrix(GRM). The GRM is built using an array of routing switches located at the intersection of horizontal and vertical routing channels. IOB and CLB architecture diagrams are available in [XilinxV]. The Xilinx CLB is comprised of four LCs(Logic Cell). An LC contains a 4-input function generator, carry logic and a storage elements. The

output from the function generator in each LC drives the D input of the flip-flop and the the CLB output. Virtex function generators are implemented as 4-input LUTs. Each LUT can provide 16x1-bit synchronous RAM. The routing is similar to previous Xilinx architectures. CLBs are interconnected to each other using local routing resources ie vertical and horizontal routing resources. The GRM(General Routing Matrix) serves as the S-Box in previous routing architectures - providing interconnection of horizontal and vertical routing resources[XilinxWeb]. Virtex is available in a 2.5V family and also a 1.8V family[XilinxWeb].

Xilinx SPARTAN for Low-Power

The Xilinx Spartan FPGA addresses the need of the low-power high voume Ubiquitous Computing market. The Spartan family is targeted to be a cheap ASIC replacement in products like - Network interface cards, portable phones, PC peripherals, badges and credit card readers. Spartan XL features on-chip RAM and is based on the Xilinx 4000 family with 5000 to 40000 system gates. The Spartan-II product is based on the Virtex-E family and extends the Spartan beyond 100000 gates. The XL family runs at 3.3 V and uses lower power than a comparable 5V device. The Spartan family is packaged in a Chip-scale package reducing the footprint of the device making it suitable for embedded applications. Spartan-II is based on Virtex-E and uses a 2.5V supply. Spartan family chips may be clocked at 200MHz and beyond.
The architecure of the Spartan-II family is shown in the Figure below along with the CLB strcuture[XilinxSpartan].

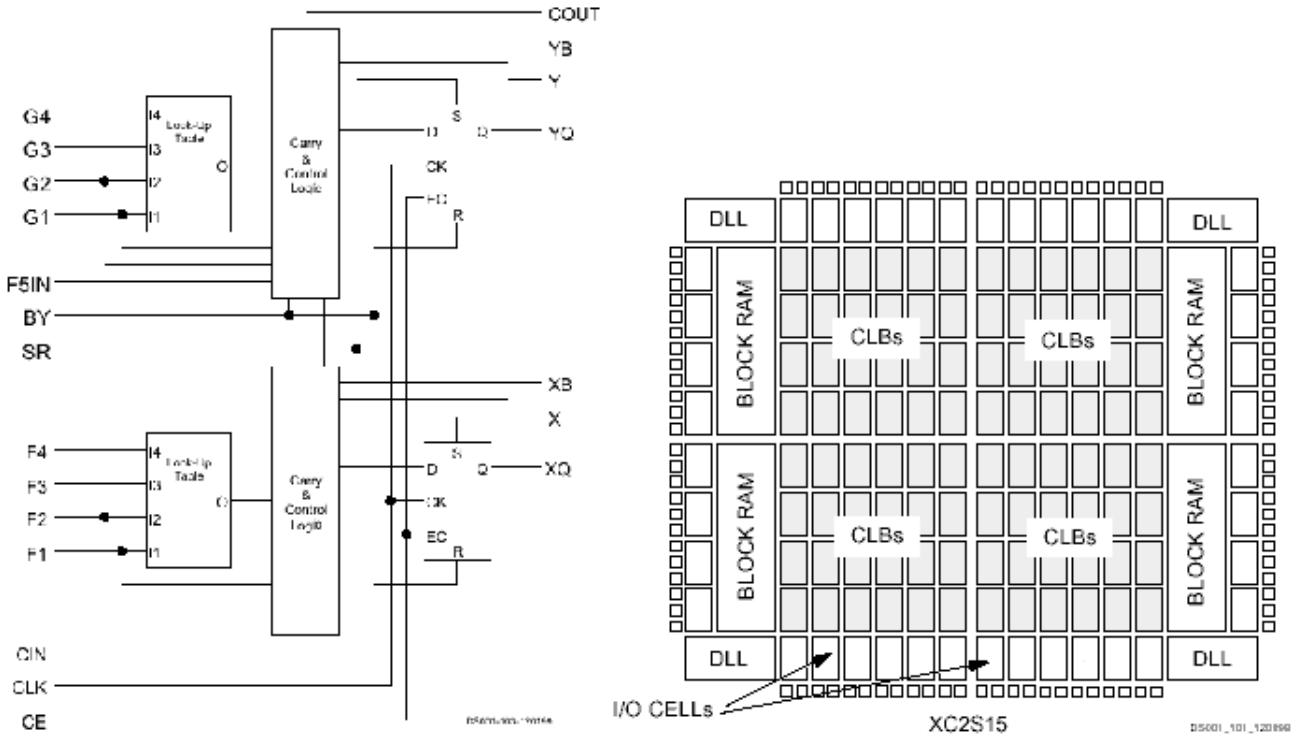


Figure 18. Xilinx Spartan CLB structure and Device Architecture[XilinxSpartan]

Spartan-XL uses a power supply of 3.3V and a power-down pin reduces the ICC to 100 microamps typical. Spartan-XL provides a manual mode where a dedicated pin is used to control power saving by

putting the Spartan-XL into a fully inactive state. In Automatic mode, where the Spartan-XL stays fully active and is completely controlled by the designer.

Spartan-II uses similar power-saving techniques as the Spartan-XL with Icc of 100 microamps and in power-down mode, all registers and memory retain their data.

Xilinx CoolRunner XPLA3 for Low-Power

Xilinx CoolRunner is a product family formed by Xilinx through a recent acquisition. The Coolrunner is a CPLD and uses a product term architecture. The family has about 750 - 9000 usable gates can be clocked from 133-200MHz for low-power embedded applications.
The pin-to-pin propogation delay is around 5ns with 3.3V and 5V operation. This CPLD is a CMOS architecture and achieves low-power by removing the sense amps usually used at the end of a product-term chain(sense-amps consume 0.25mA at Fmax). This allows 1000 times less standby current with 50% less dynamic power. ICC is less than 100 microamps[XilinxCool].

Xilinx Dynachip

During November 1999, Xilinx acquired a six-year old startup. The Dynachip FPGA during its production was the fastest clocked FPGA at 200 MHz. The product made a lot of contributions to the interconnect and routing space. The basic technology behind the company was to enhance clocking of FPGAs for high-performance applications using an active segmented interconnect. The technology developed in this startup may be used in future generations of Xilinx devices[XilinxWeb].

5.2 Altera

Altera has a product-term architecture family - MAX (CPLD) and an SRAM based family FLEX. Two new families have been added - APEX and ACE. APEX addresses the high computational density marked with millions of gates and ACE addresses the low-power embedded application space with lower gate counts but with devices that consume subnstantially lower power[AlteraApex].

APEX

Apex is unique that it allows LUT, product-term architectures and embedded memory on the same device. Traditionally, product-term architectures have been useful for state-machine decoding and other control functions and LUTs have been useful for random logic. The availability of CAM and other operations on this architecture makes it very appealing. The gates counts for this family range from 60K to 1.5 million. FIFOs, RAMs and CAMs are available on-chip. The device can be clocked at 200MHz with 2.5V and 1.8 V VDD[AlteraApex].

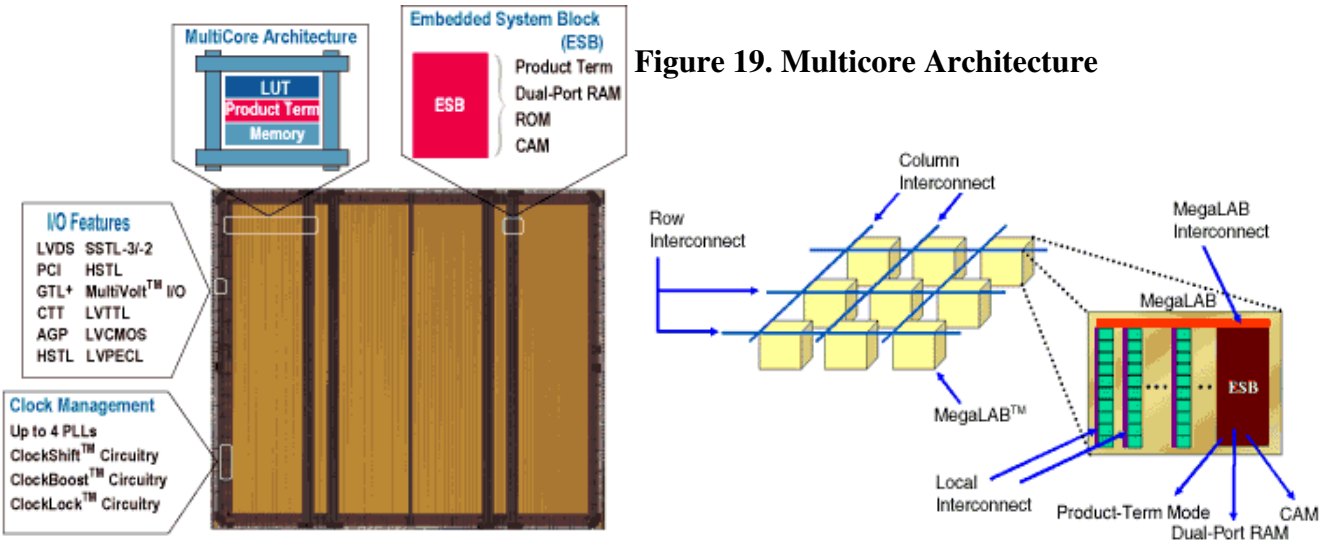


Figure 19. Multicore Architecture

the standard routing structures in previous FLEX devices with the additional use of local routing in a MegaLAB as shown in above.

ACE for Low-Power Applications

ACE is a LUT based device family that can operate at 160MHz with low power consumption. The family is based on a 1.8V 0.18-micron 6-layer metal CMOS process with gates ranging in density from 20,000 to 150,000 gates. The device family operates at 2.5 V and the datasheets report 50% power-savings[AlteraAce].

5.3 Lucent

The Lucent products are interesting because the *FPSC(Field Programmable Systems-on-a-chip)* product lines consist of heterogeneous components - FPGA and optical transceivers and PCI bus interfaces. These are intended for use in embedded applications. The FPSC is based on the ORCA3 product line. The focus in the ORCA3 product line seems to be in the direction of coase-grain logic blocks to allow dataflow style computations to be mapped to serve the needs of may telecom applications[LucentWeb].

ORCA3 FPGAs are SRAM based and consists of Programmable Logic Cells, Programmable I/O Cells and other system features. PLCs are surrounded by PICs. Each PLC contains a Programmable Function Unit, a Supplemental Logic and Interconnect and Interconnect Cell(SLIC), local routing resources and configuration RAM. Random logic can be implemented in the PFU , decoders may be implemented on the SLICs. Pre-built interfaces to i960 and PowerPC are available with a programmable Clock Manager. Orca3 is supported at the 5V, 3.3V and 2.5V levels. The number of usable gates is around 350,000[LucentWeb].

Lucent FPSC product family consists of ORCA3 components and either an optical transceiver which may be configured for SONET or
ATM framing and a PCI interface(core). The PCI core consists of independent controllers for master and

target. For the PCI interface, datapaths extend directly into the FPGA core. 120K gates are available on the device[LucentWeb].

5.4 Summary of Interesting Reconfigurable Computing Upstarts

Table 8: Reconfigurable Computing Startups

COMPANY	PRODUCT	COMMENTS
Chameleon Systems(www.chameleonsystems.com)	Systems-on-a-Chip(SOC, with FPGA)	They are targeting the stream telecom space with integratic RISC, DSP and Reprogramm logic on the same die. Very secretive.
Adaptive Silicon(www. adaptivesilicon.com)	License IP programmable logic cores for Systems-on-a-chip	All markets, mostly embedde Next-generation programmal logic cores with specified interfaces for SOC integratio
Malleable Technologies(www.malleable.com)	Build SOC chips	Telecom/embedded space wi programmable logic cores. Details sketchy. Combine po of DSP, FPGA and ASIC on same chip
Triscend(ww.triscend.com)	Build SOC reconfigurable chips(8051+fpga, ARM7+fpga)	Shipping since April 1999 w good growth roadmap
Morphics(ww.morphics.com)	Software radio with FPGA for acceleration	RISC core+DSP core+FPGA wireless market with differer standards
Cognigine(www.cognigine.com)	RISC+DSP+FPGA Systems-on-a-chip	Embedded + telecom space

---

5.5 Comments

Many academic research projects have generated new ideas and identified problems with existing reconfigurable computing architectures. Industry efforts have strived to resolve some of the issues seen and also to make reconfigurable computing more mainstream and mass-market oriented. *According to the Semiconductor Industry Association Roadmap, a single FPGA may contain more than 4 million gates.* Given the vast real-estate available, many companies are looking at placing heterogeneous components on a single die not only for effective real-estate utilization but also for providing the right components to solve a specific problem. In order for Reconfigurable Computing architectures to find more use in Ubiquitous Computing applications in the Embedded space, computational density ,

low-power and price-performance must be addressed. Also, architectures like Lucent's FPSC and National Semiconductor's NAPA 1000 will become more popular if multi-granular reconfigurable logic array(fine- and coarse-grain) are available on-chip with the software support to manually or automatically compile 'inner-loops' to reconfigurable logic arrays. A healthy cross-pollination of ideas is seen between industry and academia in the area of reconfigurable computing[HauckFuture].

6.0 Key Characteristics of Next-Generation Reconfigurable Architectures: Have the Issues been Addressed?

All the Research and Industry projects surveyed in previous sections share some common characteristics and attempt to resolve the issues listed in Section . The Table 9 below attempts to summarize the characteristics and problems addressed of both research and industry projects. The characteristics are based on the overall architecture, problems attacked or attempted to resolve and application operation space.

Table 9. Key Characteristics of Next-generation Reconfigurable Architectures

Attribute	Project[comment]
Architecture - Logic Block (new/improved)	Piperench[virtualization], DPGA[multi-context, fine-grain], Garp, Chimaera[optimized carry chains], Morphosys[multi-context, coarse-grained], Xilinx-Virtex(richer blocks), Altera(Apex)
Architecture - Systems-on-a-chip, FPSC(Field Programmable Systems-on-a-chip]	Pleiades-MAIA[SoC], Garp[Processor+RC array], Chimaera[processor+RC array], Morphosys[tinyRisc, RC array], Lucent[PCI interface, optical transceiver]
Architecture - Multiple Architectures, Data proximity	Altera-Apex[product term+LUT+memory], Xilinx Virtex[memory CAM and RAM]
Problem scope - Interconnect	Pleiades - FPGA[hierarchical, low-swing], DPGA[two-level], Xilinx-Spartan[Segmented interconnect for low-power], Altera-Apex
Problem scope - Power	Pleiades-FPGA[x40 lower than Xilinx Spartan], Pleiades-MAIA[overall pwr, low-swing], Xilinx-Spartan, Xilinx-Coolrunner, Altera-Ace
Problem scope - Configuration and Runtime reconfiguration	Piperench[runtime, static, caches], DPGA[multi-context], SCORE[virtualization, VM-style management], Chimaera[context - compression, caching, transfer, runtime reconfig], Xilinx-virtex
Problem scope - Granularity	Pleiades-MAIA[RISC, MAC, FPGA], DPGA[by context switching], Garp[processor+RC array], Morphosys[coarse-grain]
Application Space - Low-Power	Pleiades-MAIA, Pleiades-FPGA, Xilinx-Spartan, Altera-ACE
Application space - Media and DSP streams(special optimizations or considerations)	Pleiades-MAIA, Piperench, DPGA[SIMD array], Garp, SCORE, Morphosys
CAD & Compiler support(mapping, place and route)	Pleiades[templates, compilers], Piperench[compiler, mapping], DPGA[compiler], Garp[compiler - processor+rc array], SCORE[compiler+RTR software], Chimaera[compiler+high-level language compilation], Morphosys[SIMD+RC array mapping], Xilinx-virtex

7.0 Putting it All Together: The Shape of Things to Come?

Academic Research
Table 9 above shows that a number of research projects have attempted to identify many of the shortcomings in current generation systems and resolve these with new techniques or architectures. While many research groups have looked at the media stream application space, only the Pleiades project at UC Berkeley has attempted to look at low-power issues in the design of FPGAs and other reconfigurable computing architectures. While many research projects have looked at low-power synthesis, low-power VLSI CAD, only the Pleiades project has looked at the power-issue in FPGAs. Even Xilinx and Altera have attempted to reduce power only by lowering VDD and marginal improvements in interconnect. Completely new architectures are warranted to balance computational density needs with power and performance. Also, only the Pleiades project has attempted to define the applications and need for a multi-granular architecture which may be seen in future applications. There is a need for a single hardware substrate to handle both control- and data - dominated applications and only a multi-granular architecture can meet the needs of such a diverse application. The challenge is to choose the right components to meet the power budget of the application. Run-time reconfiguration is an important issue and many research projects have attempted to address many issues in this regard. It will be interesting to see how RTR may be leveraged for applications in the Ubiquitous Computing / Embedded systems space. RTR is interesting in the Ubiquitous computing space because applets or codelets from active messages may be transferred to appliances and devices and run on the system components best suited for the code. Here, important innovations in run-time compilation and just-in-time compilation may be necessary. If multi-granular architecutures are the general direction in which this community is heading then, increased support on the CAD and software side will be necessary. Applications, System software and hardware will be synthesized using Hardware/Software techniques with support for partitioning, mapping and placement on heterogeneous components of the multi-granular architecture[HauckFuture, HauckReport].

Industry Products
Table 9 also shows the representation from industry with regard to innovations in the reconfigurable computing area. The Xilinx 6200 brought in support for RTR(Run-time Reconfiguration) when single-context and multi-context designs were popular in academic and industry projects. Current industry efforts are focused on FPSC or Field-Programmable Systems-on-a-Chip where the innovations are in IP-reuse, interface between IP components and mixing different domains(analog, digital) and different processes(logic, memory like IRAM). There is currently no effort in industry directed towards new FPGA architectures for media stream processing or new architectures for low-power. It is hoped that results from academia will help fill this gap. Also, this is on account of the market realities where FPGAs are viewed as cheaper ASIC replacements and in the case of certain applications - performance acceleration. A market segment must exist if FPGAs are to be used for media streaming performance acceleration. Until then, vendors will mix domain, processes and IP components on a single chip to meet the needs of applications with different computational requirements[HauckFuture, HauckReport].

Technology and Hardware Trends
Technology and market trends affect the growth of an area. FPGAs are currently popular for logic emulation and prototyping and as cheap ASIC replacements. FPGAs are slowly invading the Mask-programmable Gate Array market while the low-end FPGA market is being eroded by PLDs and embedded processors. FPGAs are currently supporting 300,000 gate designs on a chip. *Based on data from vendors and the Semiconductor Industry Association Roadmap, it is expected by 2010 that there will be around 4 million gates on a single chip.* This capacity clearly needs to be utilized in an effective

manner (currently 90% of the chip area is interconnect) using innovations in interconnect design such as Hierarchical interconnects and using active instead of passive interconnects. Also, since VLIW processors (*http://www.transmeta.com*) are gaining popularity for media applications, it may also make sense to include VLIW cores on SoC solutions that can handle both control- and data- dominated workloads. In fact, VLIW processors with the right compiler support are being seen as replacements for DSPs which have optimized and high throughput multiply units. Also, the low-power application space needs to be addressed in a more directed fashion. Most products and research are geared towards increasing computational density and high-performance but it is also necessary to make power-performance tradeoffs to serve the need of reconfigurable computing applications in the low-power embedded computing space where performance, power, battery-life, density and cost must be considered in an integrated fashion. VLSI fabrication processes must adapt to allow mixing of different domains (analog and digital) and also different processes(logic and memory) on the same die to allow realization of Field Programmable Systems-on-a-Chip solutions[HauckFuture, HauckReport].

CAD and Compiler Trends

Clearly, support from CAD and compilers is critical for the success of Reconfigurable computing in all application domains. Research in Hardware/Software Codesign will allow compilation from high-level languages will allow a wider developer base to use the hardware and design tools. Synthesis techniques must include low-power optimizations to serve the need of the Ubiquitous Computing/Embedded systems space. Also, in the case of SoC and FPSC solutions, automatic partitioning and placement across components of different granularity will be needed. Support for configuration management will require a combined hardware/software approach with possible support from a run-time compiler for relocating mappings(new symmetrical architecture needed here) and allowing partial run-time reconfiguration. Also, development of Reconfigurable SoC and FPSC solutions may benefit from template-based design methodologies like those advocated by the Pleiades group[UC96] and companies like Tensilica(*http://www.tensilica.com*)[HauckFuture, HauckReport].

Newer Applications

Random logic(for fine-grained FPGAs), datapath-style operations(for coarse-grain FPGAs) have allowed numerous stream media computations to be implemented on FPGAs with remarkable speedups. These are found in image processing kernels and other DSP kernels. Streams and random logic are good candidates for implementation on FPGAs. For mass-market appeal of FPGAs, it is important for the research community to look at newer applications. In one application, Adobe Photoshop was accelerated using a Xilinx 6200 FPGA card. With security becoming an important application, it may be interesting to implement a number of security algorithms in FPGAs. Also, with standards changing very rapidly in the wireless and wireline protocol world, FPGAs can be programmed in-situ to adapt to the changed protocol standard[HauckFuture, HauckReport].

Academic Research+Industry projects+Technology trends+CAD/Compiler Trends+Newer Applications = FPSC ?

Field programmable Systems-on-a-Chip(FPSC) seems to be the natural result of a culmination of different forces - increasing chip real-estate, better CAD and compiler support and mixing of different processes and domains. Such integration will serve the needs of applications with multi-granular computational needs and a mix of control- and data - dominated workloads[HauckFuture, HauckReport].

These are expected to contain the following components -

- VLIW or RISC processor
- Reconfigurable Array(Mult-granular, fine grained and coarse-grained structures, possibly arranged in SIMD fashion)
- DSP
- DRAM cells(If mixing of DRAM memory processes and logic becomes viable for large amounts of memory)
- Interconnection network(With appropriate hierarchical routing)
- Mixed-signal interfaces to the analog world

8.0 Conclusion

The strengths and shortcomings of current reconfigurable architectures have been explored. The need for increased use of FPGAs in Ubiquitous Computing environments has been justified. A number of Next-generation reconfigurable architectures in academia and industry have been surveyed and described. The key characteristics of these architectures have been extracted and presented in a tabular fashion. Technology trends have been identified that may result in the use of Reconfigurable Systems-on-a-Chip solutions.

A hard question to answer is the course of events in the 'generation-after-next'. If fine-grained reconfigurations are possible in hardware then, it may be evolved to adapt with the performance needs of the application - some kind of performance annealing. Fusion of genetic algorithms and AI research may open doors into hardware that may change and evolve over time - the exciting realm of Evolvable Hardware[Evolve].

References

[UC96] Arthur Abnous and Jan Rabaey, "Ultra-Low-Power Domain-Specific Multimedia Processors," Proceedings of the IEEE VLSI Signal Processing Workshop, San Francisco, California, USA, October 1996.
[UC97]Jan M. Rabaey, "Reconfigurable Computing: The Solution to Low Power Programmable DSP," Proceedings 1997 ICASSP Conference, Munich, April 1997.
[UC98] Hui Zhang and Jan Rabaey, "Low-Swing Interconnect Interface Circuits", ISLPED 1998
[UC99]Varghese George, Hui Zhang, Jan Rabaey, "Design of a Low Energy FPGA", ISLPED 1999
[UC2000] Hui Zhang, Vandana Prabhu, Varghese George, Marlene Wan, Martin Benes, Arthur Abnous, "A 1V Heterogeneous Reconfigurable Processor IC for Baseband Wirelss Applications", Proceedings of ISSCC2000.
[UCplweb] The Pleiades Group Web page at UC-Berkeley
http://bwrc.eecs.berkeley.edu/Research/Configurable_Architectures/

[CMU99]Seth Copen Goldstein, Herman Schmit, Matthew Moe, Mihai Budiu, Srihari Cadambi, R. Reed Taylor, Ronald Laufer "PipeRench: a Coprocessor for Streaming Multimedia Acceleration" in ISCA 1999.
[CMU98]Srihari Cadambi, Jeffrey Weener, Seth Copen Goldstein, Herman Schmit, Donald E. Thomas, "Managing Pipeline-Reconfigurable FPGAs," in Proceedings ACM/SIGDA Sixth International Symposium on Field Programmable Gate Arrays, Feb. 1998.
[CMU97]Herman Schmit, "Incremental Reconfiguration for Pipelined Applications," in Proceedings of the IEEE Symposium on FPGAs for Custom Computing Machines, pp. 47-55, 1997.
[CMUweb] The PipeRench Web page at CMU - http://www.ece.cmu.edu/research/piperench/

[DPGA] Andre Dehon, "Dynamically Programmable Gate Arrays: A Step Towards Increased Computational Density", in Proceedings of Fourth Canadian Workshop on Field Programmable Devices, May 13-14, 1996, Toronto, Canada
[DPGAweb] MIT Reinventing Computing, DPGA page - http://www.ai.mit.edu/projects/transit/dpga_prototype_documents.html

[UCGarp] ‘‘Garp: A MIPS Processor with a Reconfigurable Coprocessor,’’ by John R. Hauser and John Wawrzynek, published in Proceedings of the IEEE Symposium on Field-Programmable Custom Computing Machines (FCCM ’97, April 16-18, 1997).
[UCScore] UC Berkeley SCORE project web page - http://brass.cs.berkeley.edu/SCORE/

[NWU]S. Hauck, T. W. Fry, M. M. Hosler, J. P. Kao, "The Chimaera Reconfigurable Functional Unit", IEEE Symposium on FPGAs for Custom Computing Machines, pp. 87-96, 1997.
[NWUweb] Chimaera Adaptive Computing - http://www.ece.nwu.edu/~hauck/DARPA/index.html

[UCI] Nader Bagherzadeh, Fadi J Kurdahi, Hartej Singh, Guang-ming Lu and Ming-Hau Lee, "Design and Implementation of the MorphoSys Reconfigurable Computing Processor " , to be published in Journal of VLSI and Signal Processing-Systems for Signal, Image and Video Technology, March 2000
[UCIweb] Morphosys Project Site at UC Irvine - http://www.eng.uci.edu/morphosys/morphosys-new/

[XilinxWeb] http://www.xilinx.com
[XilinxSpartan] http://www.xilinx.com/products/spartan2/arch.htm
[XilinxCool] http://www.xilinx.com/products/xpla3.htm#xpla3family

[AlteraAce]http://www.altera.com/html/products/ace.html
[AlteraApex]http://www.altera.com/html/products/apex.html

[LucentWeb] http://www.lucent.com/micro/fpga/

[HauckFuture] S. Hauck, "The Future of Reconfigurable Systems" , Keynote Address, 5th Canadian Conference on Field Programmable Devices, Montreal, June 1998.

[HauckReport]S. Hauck, "The Roles of FPGAs in Reprogrammable Systems" , Proceedings of the IEEE, Vol. 86, No. 4, pp.
615-638, April, 1998.
[Evolve] Evolvable Hardware page - http://www.etl.go.jp/~ehw/eng/frame/main.html